Batched Reproducible and Reduced Precision BLAS Forum – SC’18

Presented by
Siva Rajamanickam

Kyungjoo Kim, Vinh Dang, Andrew Bradley, Micah Howard, Sandia National Laboratories
Compact BLAS in Kokkos Kernels update
Kokkos Ecosystem for Performance Portability

Kokkos Core: parallel patterns and data structures, supports several execution and memory spaces

Kokkos Kernels: performance portable BLAS, sparse, and graph algorithms and kernels

Kokkos Tools: debugging and profiling support

Kokkos Ecosystem addresses complexity of supporting numerous many/multi-core architectures that are central to new Supercomputers
Kokkos Kernels Compact BLAS / Block Interleaved format

- Data layouts are critical for performance
  - Divide the batch of matrices in blocks
  - Interleave the blocks
  - Block enough to fill vector units and fit in cache
- Compact layout: Block size is same as vector length
- Pros for compact layout: Improves vectorization
- Cons for compact layout: Might need repacking the data for new layouts
- See SC’17 paper for performance comparisons

Standard data layout

Compact data layout using vector length of 4
One Year later …

- Developed new preconditioners based on Compact BLAS Kernels
- Integrated the compact BLAS based preconditioners into production application (SPARC) that is part of the Exascale Computing Program and ATDM program
- Continued work with Intel Compact BLAS
- Home grown version of Compact BLAS on the GPUs as required by the applications
  - Performance can be improved with vendor support (hint, hint ..)
- Support for several new kernels getri, getrs,
- Working on other new kernel QR, Hessenberg form, Eigen solvers
Next Steps

- Continue to develop team level compact kernels
- Contrary to Batched BLAS community’s original plans most popular application need is the compact BLAS over traditional way of calling batched BLAS
  - Applications like control of “parallel for” and call BLAS within the loop (teams)
  - Use cases different – minimal error checking, vectorization needs high
  - Use cases for variable block size growing
  - [https://github.com/kokkos/kokkos-kernels/issues/9](https://github.com/kokkos/kokkos-kernels/issues/9)
- Some of the above use cases are addressed in some implementations
  - No complete set
  - GPU support for compact BLAS is lacking
- Standardization of the C++ version of the reference, with team level interface, would be nice
- One document for C, C++, team level and device level interface would be nice